



PCT/AU03/00900

23

Rec'd PCT/PTO 11 JAN 2005

**PRIORITY  
DOCUMENT**

SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

Patent Office  
Canberra

REC'D 31 JUL 2003

WIPO PCT

I, JULIE BILLINGSLEY, TEAM LEADER EXAMINATION SUPPORT AND  
SALES hereby certify that annexed is a true copy of the Provisional specification  
in connection with Application No. 2002950122 for a patent by WEBND  
TECHNOLOGIES. as filed on 11 July 2002.



WITNESS my hand this  
Twenty-second day of July 2003

*J. Billingsley*

JULIE BILLINGSLEY  
TEAM LEADER EXAMINATION  
SUPPORT AND SALES

BEST AVAILABLE COPY

**AUSTRALIA**

*Patents Act 1990*

**COMPLETE SPECIFICATION**

**PROVISIONAL PATENT**

**Software process for management by non-technical  
users of electronic page design, content and layout in  
a distributed environment**

**The invention is described in the following statement:**

**BEST AVAILABLE COPY**

## **Software process for management by non-technical users of electronic page design, content and layout in a distributed environment**

For many administrators the inability to control the structure of a page without programming skills creates an undesirable dependence on programmers. Various systems exist for managing content objects on electronic pages, but each uses hard coding to define where content objects appear on a page. Using hard coded pages or templates to define where dynamic content appears creates a dependence on administrators with programming skills and restricts the scope of non-technical administrators to manage page design, content and layout.

By using a hard coded page or template to define a page structure and layout other systems essentially limit themselves to two levels of structure, namely the page level, and the content level. Existing products require a programmer to create this basic page structure using one or more tables or other similar devices. Only the content level can be delegated to a non-programmer, because control over the page level requires programming skills.

Specifically, existing systems have the following three limitations:

1. Creating new page structures requires programming skills
2. To manage web page design, content and layout or share structural objects of a page requires programming skills
3. Administration of structural objects of a page cannot be delegated

Page structure starts with a rectangular page and uses subdivisions to create smaller rectangular subsections of the page. By nesting tables and similar devices in different combinations an endless variety of page structures can be created. Traditionally programmers are required to create the page structures, but using the Web<sup>ND</sup> System the same structure can be created from scratch by without any programming skills, overcoming previous limitations and enabling non-technical users to create and manage page design, content and layout in a distributed environment.

The Web<sup>ND</sup> System utilizes a unique system of pointers (tags), content pools (pools) and nested table structures. A Web<sup>ND</sup> page is a single container into which is inserted a content pool containing an unlimited number of containers. Each of these containers can hold and format another content pool, which can in turn hold an unlimited number of containers. This process can be repeated indefinitely to create a structure with sufficient levels of nested content containers organized appropriately into content pools. By using a wizard to define appropriate table layouts in which to wrap content containers from content groups, the desired page structure can be achieved. With the desired page structure in place containers within the structure are easily utilized to create the look and feel of the page and manage content.

Each container and pool can be identified and displayed independently. This enables subsections of a page (including columns, rows, and other page elements) to be managed individually and displayed in multiple locations. By assigning permissions to modify specific content items non-programmers can add page structure as well as content to any part of the page and manage it easily. This system divides the traditional page level of control into many distinct levels of control, each of which can be delegated to non-programmers. Like traditional systems, the content level can

**BEST AVAILABLE COPY**

also be delegated to non-programmers. The levels of control created are explained below.

#### **Level 1**

The first level of a pages control structure is the definition of the page itself. No html is coded into WebND pages. Each page is defined instead by data stored in a database, which includes:

- A unique Page ID number
- A Page Title
- A single Tag ID number

#### **Level 2**

The Tag associated with the page defines the next level of the pages control structure. No HTML is coded into Tags themselves. Each Tag is defined instead by data stored in databases which includes:

- A unique Tag ID
- A Tag Title (if applicable)
- Additional display instructions (if applicable)

and either:

- The location of the raw content (if the Tag is used to display content), or
- A single Pool ID number (if the Tag is used to create the page structure)

If the Tag is used only to display content, the content simply appears in place of the Tag when ever the page is displayed. If, however, the Tag is used to build upon the existing page control structures, it points to a single content Pool.

#### **Level 3**

The Pool pointed to by the Tag defines an additional level of the pages control structure. No HTML is coded into the Pool itself. Each Pool is defined instead by data stored in databases which includes:

- A unique Pool ID number
- A Pool Title (if applicable)
- A list of Tags ID numbers and their associated rank

The Pool can contain any number of Tags each with its own rank within the Pool. The display instructions (noted in Level 2 above) can use a simple table to display items in the Pool with one Tag displayed in each cell of the table. This enables nonprogrammers to control the page layout by defining the number of columns to use. For example, to divide the page into three columns (left, middle and right), the display instruction setting for number of columns is set to three and the appropriate coding is included automatically when the page is displayed.

#### **Additional Levels**

The Tags associated with the Pool at Level 3 define a further level of the pages control structure. All Tags have the same structure and are stored in the same database. Specifically, the Tag used at Level 1 is the same as the Tags in the Pool at level 3. Any Tag can point to a Pool. This Pool can in turn contain Tags which themselves point to Pools, and the pattern can be repeated indefinitely until the desired page structure is achieved.

To assist with understanding the invention, reference will now be made to the accompanying drawings that show one example of the invention.

**BEST AVAILABLE COPY**

In the drawings:

FIG. 1 shows one example page structure with a header, footer and two columns in the body, which are all touching, as they would be when displayed.

FIG. 2 shows one example of the use invention to create the example page structure.

Referring to FIG. 1 an example page structure can be seen which includes a Header 1, a left Content Area 2, a right Content Area 3 and a Footer 4. The page layout as seen is formed using a structure of nested control devices, but such devices are not usually visible on the rendered page.

Referring to FIG. 2 the example page can be seen with the various structural control devices shown symbolically separated by white space so their contribution to the page structure can be illustrated.

The Page 1 is shown to contain a Tag 2 that in turn contains a Pool 3.

The Pool 3 can be seen to contain a Table 4 comprising of three cells arranged in one column with three rows.

The Tags 5, 6 & 7 contained in the Pool 3 can be seen inside the table cells with one Tag appearing in each cell effectively dividing the Page 1 into three rows. The Tag 5 in the top row of the Table 4 can be seen to contain Content 8 that forms the header. The Tag 7 in the bottom row of the Table 4 can be seen to contain Content 9 that forms the footer. The Tag 6 in the middle row of the Table 4 can be seen to contain more levels of structural control devices.

The Tag 6 in the middle row of the Table 4 can be seen to contain a Pool 10. The Pool 10 can be seen to contain a Table 11 comprising of two cells arranged as two columns in one row.

The Tags 12 & 13 contained in the Pool 10 can be seen inside the Table 11 with one Tag 12 seen in left cell and the other Tag 13 seen in the right cell. The two Tags 12 & 13 can be seen to contain content that forms the left and right columns of the body of the Page 1.

#### **Natural extensions included in the invention**

By not defining the number of columns a table uses when it wraps tags contained in a pool to form columns or rows on a page, the number of tags can determine the number of columns to be displayed instead. This enables additional rows or columns to be added by simply adding a new tag to the pool. Similarly, columns can be removed by removing a tag from a pool.

Modifying the ranks of the tags within the pool can change the order in which related columns and rows are displayed. By Defining additional ranking criterion to tags within a pool additional the control over the order the number of tags displayed can be achieved. This enables greater flexibility to define the order in which columns and rows are displayed. Variables that are changing, random, or even external to the invention can be incorporated to influence page structure.

**BEST AVAILABLE COPY**

When the number of tags displayed is greater than the number of columns, by simple calculation the table can be automatically adjusted to incorporate additional rows.

By giving all structural objects, including tags and pools, identification numbers and linking them to users via permissions tables we created a system whereby administration of any object is easily delegated and managed.

By presenting the same page structural objects in modified formats, customized administration and powerful management interfaces can be utilized.

Page structural objects can use templates to achieve many benefits including:  
Improved efficiency; enhanced appearance; consistent look and feel.

**BEST AVAILABLE COPY**

FIGURE 1

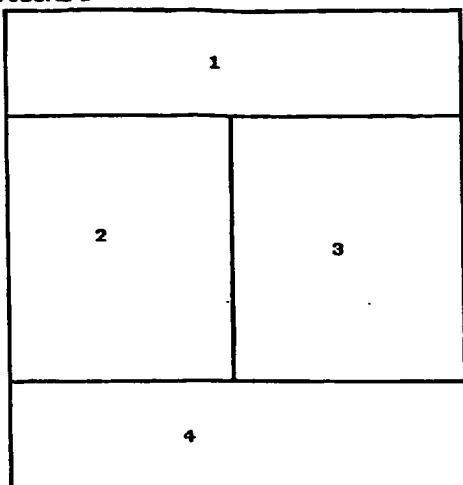
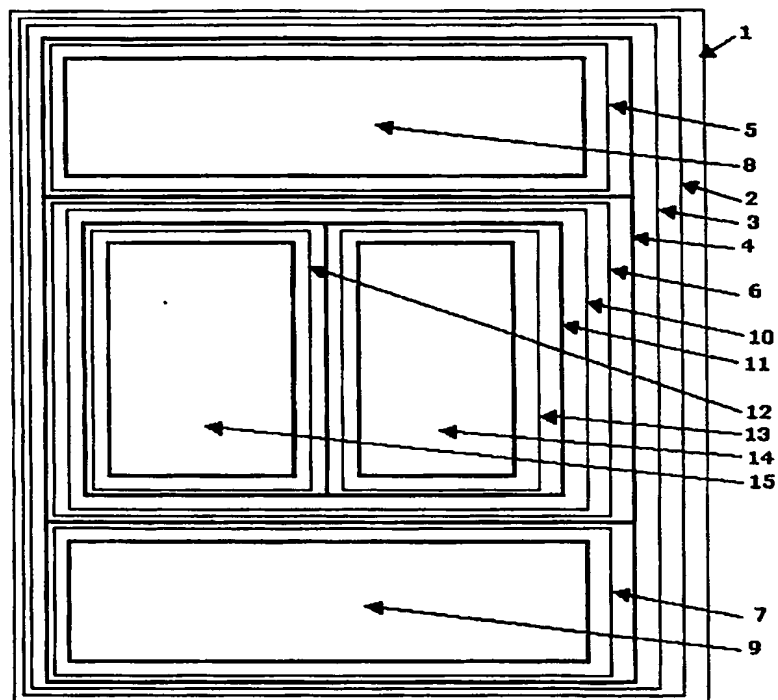


FIGURE 2



BEST AVAILABLE COPY